Predicting Online Sales from Webpage Analytics

Lesley Miller

24/01/2020

Contents

Project Summary

Here we attempt to build a classification model using the light gradient boosting algorithm which can use webpage metrics from a given online shopping website to predict whether the revenue of a new customer is True (i.e., the customer purchased something) or False (i.e., the customer did not purchase anything). Our final classifier performed well on an unseen test data set, with the f1 score of more than 0.6 and the test accuray of ~90%. The precision and recall of our classifier on the test set are also around 0.6. However, because of the small target frequency we have a high percentage of incorrect prediction in FN, we plan to further investigate & improve our model.

Introduction

Online shopping has quckly become a dominant player in commerce. It's been reported that within the next 3 years, 91% of those in the United States will have shopped online! That is nearly 300 million people for the US alone. Additionally, in 2020 the prediction is that 4 trillion dollars will be spent by online shoppers. Given this enormous potential, online retailers want to know reliable ways to predict user behavior and uncover insight into what factors are most predictive of sales. The following analysis builds a binary classifier to be able to predict a sale coded as `TRUE` or no sale coded as `FALSE`.

Data Source

The data set used in this project is of online shopping webpage metrics created by C. Okan Sakar, S. Olcay Polat, Mete Katircioglu & Yomi Kastro(Sakar et al. 2019). It was sourced from the UCI Machine Learning Repository(Dua and Graff 2019) and can be found here.Each row in the data set represents webpage metrics on a single shopper which was extracted from the URL information and includes the final action (purchase or not) and several other measurements (e.g., Number of Distinct Product Related pages, Time spent on Product Related pages, closeness of site visitng time to a special day, etc.).

Exploratory Data Analysis

Each row in the dataset represents a session by a single user with a total of 12,330 shopping sessions. Each unique user is represented only once in the entire dataset and is form a 1-year period. A total of 15% of the sessions end in a purchase. The table below describes each of the 18 measurements made on each online shopper.

- Each row represent a session by a user.
- Each user has only 1 session in the dataset.
- The data is for 1-year period.
- ~15% sessions resulted in a purchase.
- Predictive variables included in this analysis are user's visit information, web analytics features & geographic features.

Description of the variables, data source

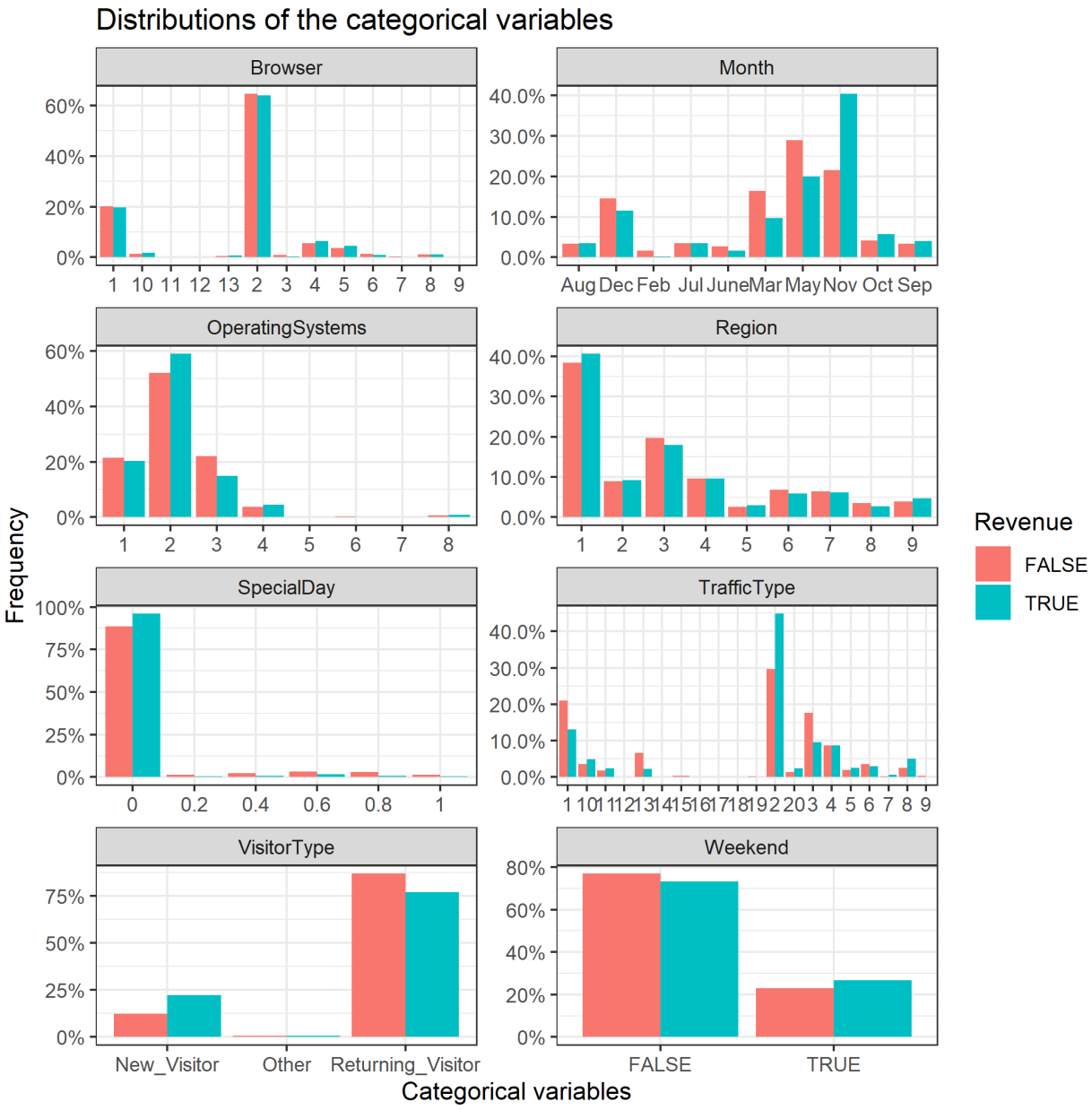
No.	Variable	Description
1	`Administrative`	Number of Distinct administrative pages
2	`Informational`	Number of Distinct Informational pages
3	`ProductRelated`	Number of Distinct Product Related pages
4	`Administrative_Duration`	Time(in seconds) spent on Administrative pages
5	`Informational_Duration`	Time(in seconds) spent on Informational pages
6	`ProductRelated_Duration`	Time(in seconds) spent on Product Related pages
7	`BounceRates`	Average bounce rate of all web-pages visited by user. For a web-page its the percentage of people who visit the website from that webpage and left without raising any other request
8	`ExitRates`	Average exit rate of all web-pages visited by user: For a web-page its the percentage of people who exited the website from that webpage
9	`PageValues`	Average page value of all web-pages visited by user: For a web-page its the average dollar-value of that page which the user visited before completing the transaction
10	`SpecialDay`	The closeness of site visitng time to a special day (higher chances of a session resulting in a transaction)

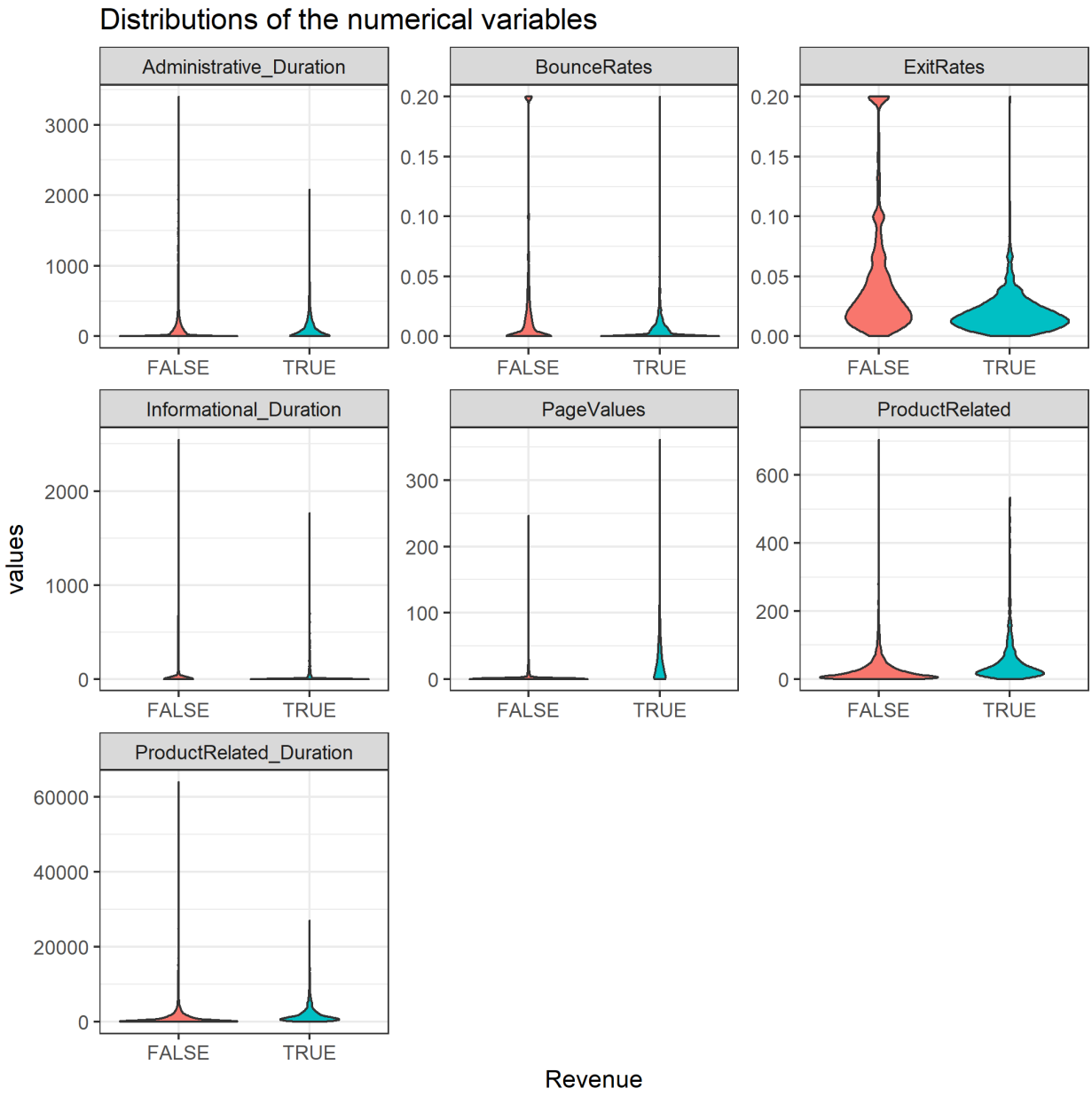
Table 2: Cumulative Distribution of Numberic Variables

probs	Administrative_Duration	Informational_Duration	ProductRelated_Duration	BounceRates	ExitRates	P
0.00	0.000	0.000	0.000	0.000	0.000	
0.05	0.000	0.000	0.000	0.000	0.005	
0.10	0.000	0.000	36.000	0.000	0.007	
0.15	0.000	0.000	77.000	0.000	0.010	
0.20	0.000	0.000	128.000	0.000	0.012	
0.25	0.000	0.000	181.500	0.000	0.014	
0.30	0.000	0.000	243.983	0.000	0.016	
0.35	0.000	0.000	315.000	0.000	0.018	
0.40	0.000	0.000	402.009	0.000	0.020	
0.45	0.000	0.000	493.640	0.000	0.023	
0.50	8.000	0.000	598.600	0.003	0.025	
0.55	22.971	0.000	718.855	0.005	0.029	
0.60	38.000	0.000	855.133	0.007	0.032	
0.65	54.500	0.000	1029.246	0.010	0.036	
0.70	73.020	0.000	1229.497	0.013	0.041	
0.75	95.763	0.000	1473.834	0.017	0.050	
0.80	124.640	0.000	1784.082	0.023	0.058	
0.85	165.388	25.185	2210.075	0.033	0.073	
0.90	228.320	70.000	2905.811	0.059	0.100	
0.95	354.752	193.463	4322.162	0.200	0.200	
1.00	3398.750	2549.375	63973.522	0.200	0.200	

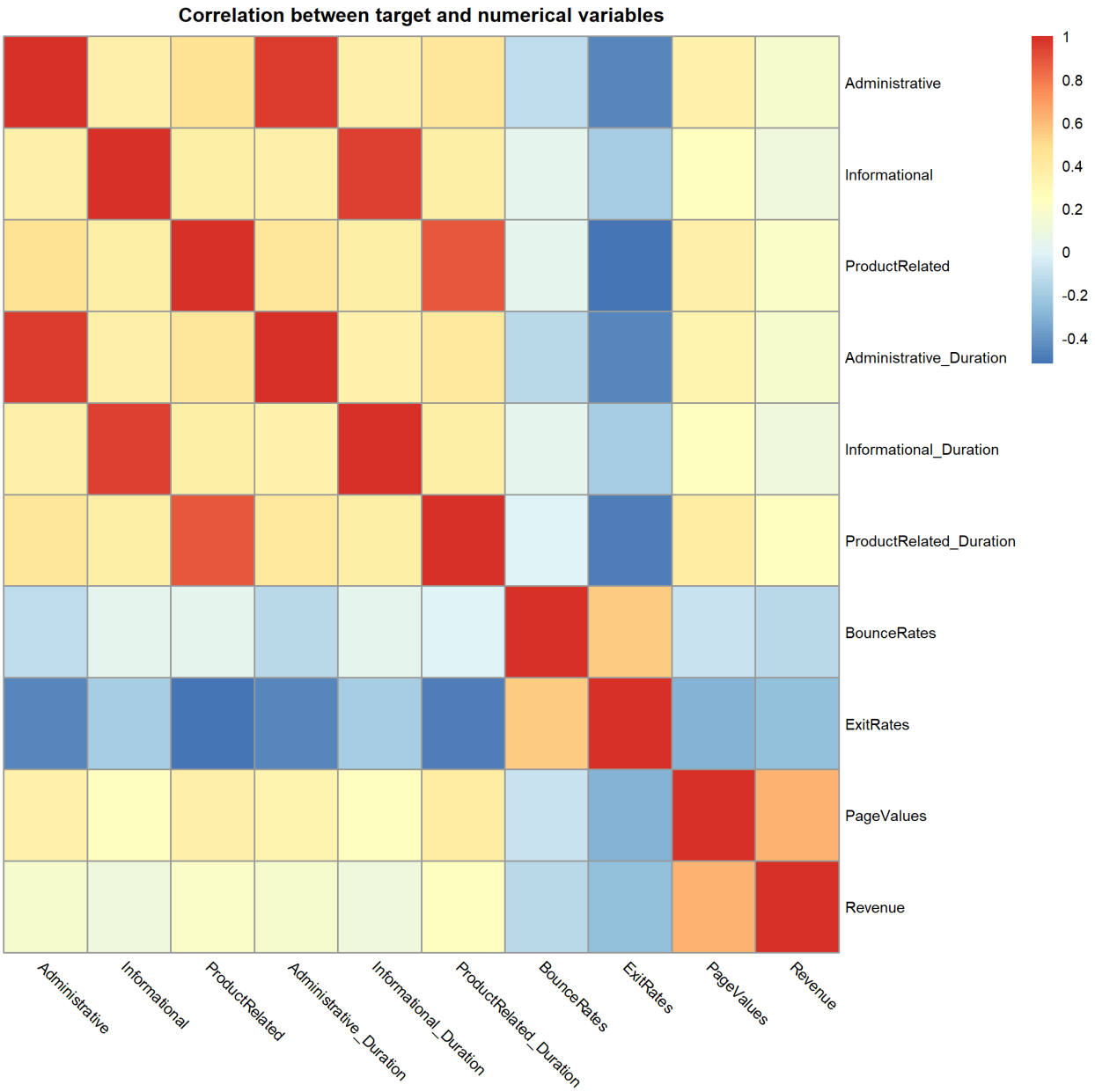
No.	Variable	Description
11	`OperatingSystems`	Operating system used by the user
12	`Month`	Month of Year
13	`Browser`	Browser used by the user
14	`Region`	Geographic region
15	`TrafficType`	Type of Channel user by the user to arrive at the website
16	`VisitorType`	Type of the visitor
17	`Weekend`	Weekend indicator
18	`Revenue`	Revenue transaction indicator

The multi-panel plot below details the distributions of the 8 categorical variables. It highlights how several variables are quite skewed, since a in a given category, its values tend to be clustered around only a few values. For example, there is clearly one browser that dominates all the others regardless if there is a sale or not. The months of March, May and November constitute a substantial share of the dataset. Also with different operating systems, 4 are the most frequently used while the other 4 are barely represented.


Distributions of the categorical variables
Browser
Month
OperatingSystems
Region
SpecialDay
TrafficType
VisitorType
Weekend
Frequency
Categorical variables
Revenue
FALSE
TRUE
New_Visitor
Other
Returning_Visitor
Aug Dec Feb Jul June Mar May Nov Oct Sep



We ran correlation analysis to see how the predictor variables are correlated with each other as well as see how the predictors are related linearly to the outcome, `Revenue`. The heathmap below visualizes the correlation. The most positively correlated are orange to dark red while the negatively correlated variables are in progressively dark blue. Several variables are either not correlated or only weakly correlated. But others ariables, like the page type `Administrative` is strongly correlated with the the time spent on the `Administrative` page coded as `Administrative_Duration`. This is also true of the other page types, `Informational` and `Product Related`. These too, are highly correlated with the time spent on them which intuitively makes sense. `PageValues` which is the average dollar value of the page, is the most positively correlated with `Revenue`; while `Exit Rates` and `Bounce Rates` have slight negative correlation with `Revenue`.



Modelling

We are trying to predict the chances of online purchase by a customer. Since this is a classification problem we tried Machine Learning models that are good for classification and tested their accuracy by optimizing the hyperparameters. We mainly tried Python(Van Rossum and Drake 2009)'s Scikit Learn(Pedregosa et al. 2011)'s Logistic Regression(`sklearn.linear_model.LogisticRegression()`), Random Forest(`sklearn.ensemble.RandomForest()`) & Light Gradient Boosting(`lightgbm.LGBMClassifier()`) Technique. The best accuracy on the validation data was achieved by `LGBM`, therefore we went ahead with automating the entire analysis using `LGBM` only.

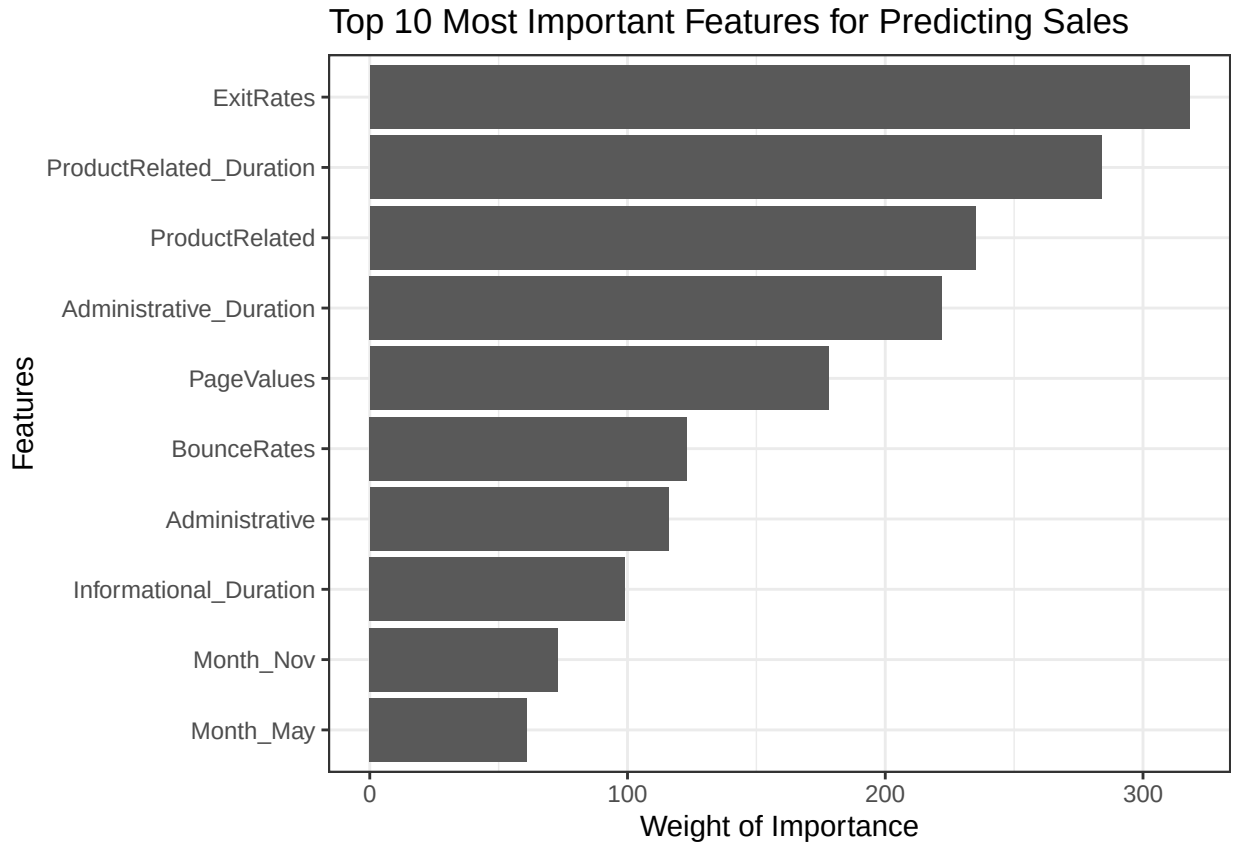
During Exploratory Data Analysis(EDA), we used R(R Core Team 2019) and found that the customers who made an online purchase were a small fraction of all the users (~15%). Due to this, there was a possibility of facing the class imbalance issue. Because of this, we included the oversampling strategy(`None`: No oversampling, `balanced`:automatically balancing the weights inversely propotional to the class frequencies)

Table 3: Best hyperparameters chosen

Hyperparameter	Value
class_weight	NA
max_depth	5
n_estimators	100

provided by `LGBM` in our hyper-parameter optimization and the modeling framework to automatically decide the most optimal oversampling strategy.

From the initial EDA, we also saw that the variables had a highly skewed distribution with many outliers. Due to this challenge, we preprocessed the data using Quantile Transformation (`sklearn.preprocessing.QuantileTransformation()`) provided by Scikit Learn. This method transforms the individual explanatory variables to follow normal distribution. This transformation is a robust preprocessing technique and reduces the impact of outliers present in the data.



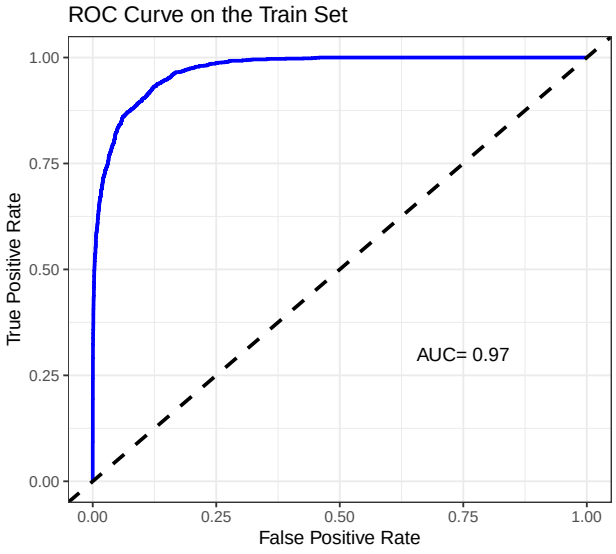
The best hyperparameters after grid search is shown below:

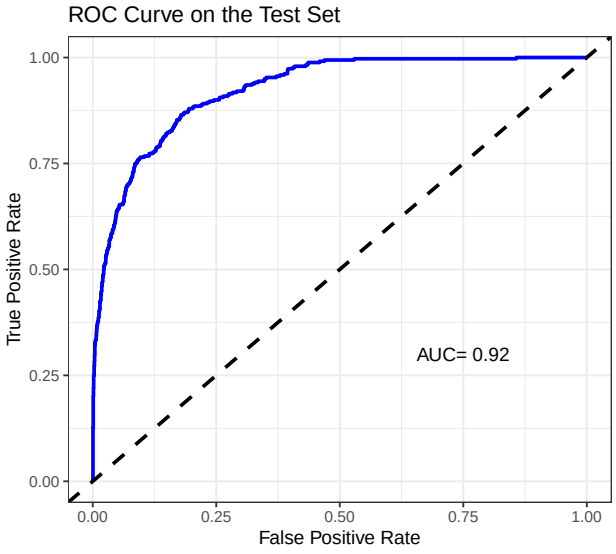
Results and Discussion

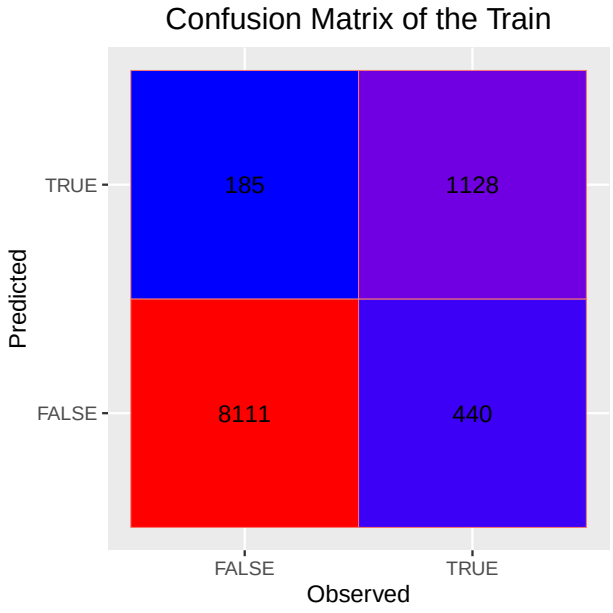
For evaluating the performance of the classification models we are using the following metrices:
- **ROC-AUC Curve**: Area Under the ROC curve is an apt measure of the measuring the performance of the model because it doesn't depend on the threshold chosen for classifying the predicted class.
- **Confusion Matrix**: Based on the chosen threshold this matrix highlights the classification(True Positives & True Negatives) & misclassification(False Positives & False Negatives) done by the model. The threshold

chosen for this matrix is automatically decided by the model based on maximum F1-Score. By looking at the confusion matrix we can know the precision, recall & F1-Score of the predictions which gives a holistic view of the performance of the model.

- **Effect of Threshold on Performance**: Deciding the threshold of the predicted probability for classification requires human intervention because sometimes the objective is not only maximizing the accuracy. For e.g. a business might want to decide a threshold that minimizes their cost which is a function of (True/False Positive/Negative). Therefore, we are producing this output which highlights how the performance of the model will change by changing the threshold which can help the stakeholders decide on the optimum threshold.







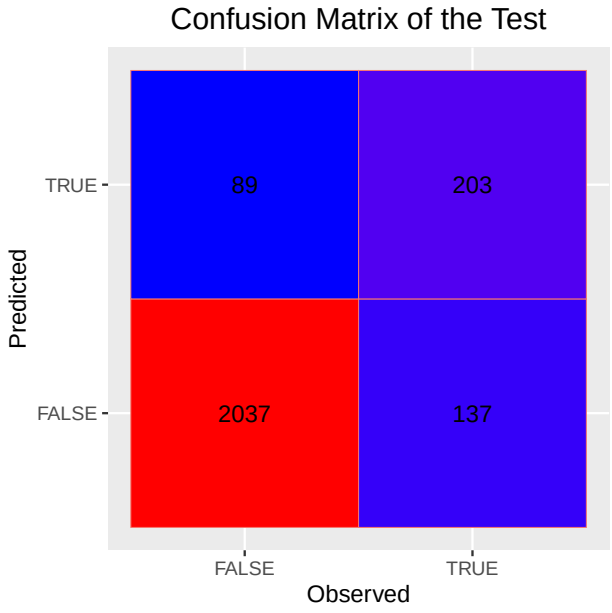
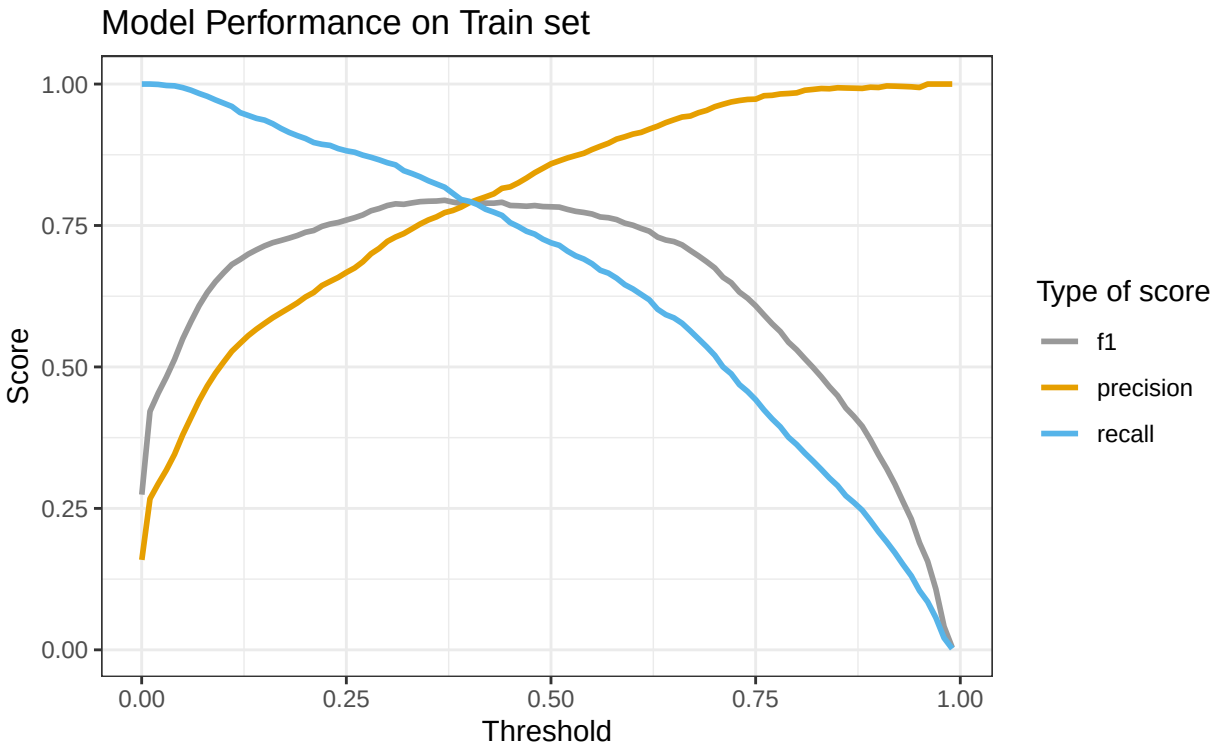
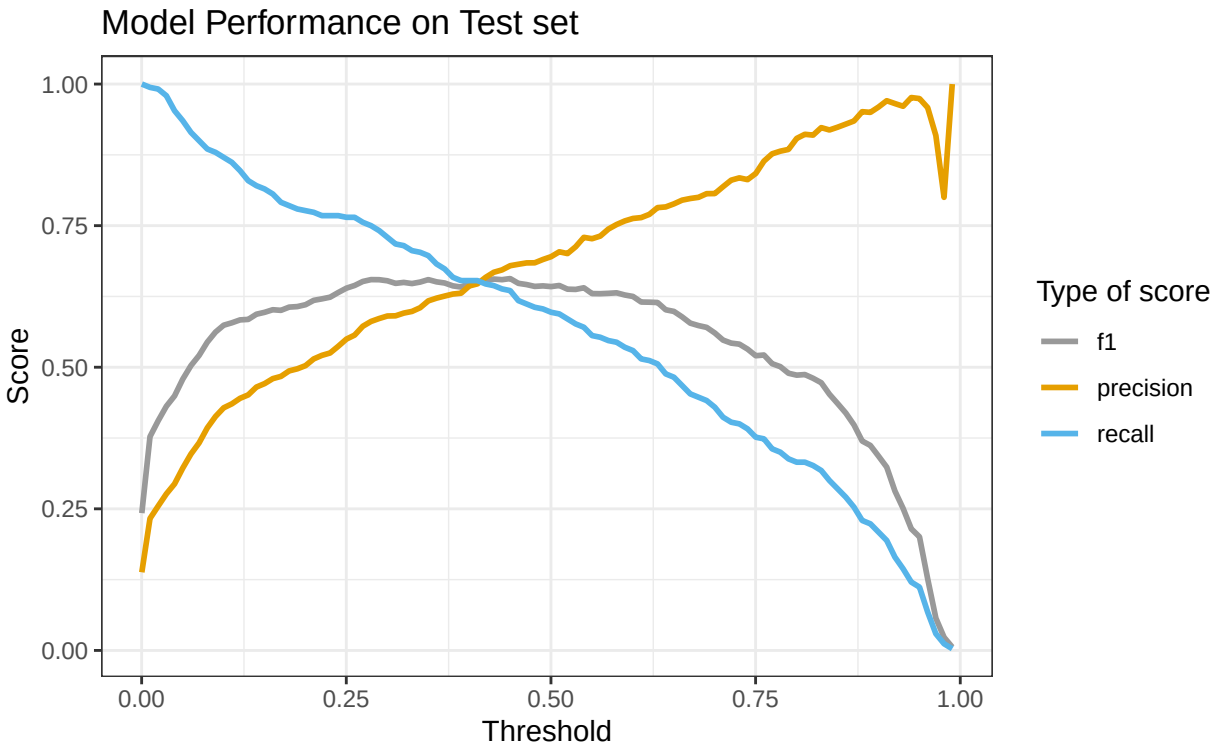


Table 4: Model Evaluation Metrics on Train

Dataset	Precision	Recall	F1_Score
Train	0.859	0.719	0.783
Test	0.695	0.597	0.642





References

Dua, Dheeru, and Casey Graff. 2019. "UCI Machine Learning Repository." University of California, Irvine, School of Information; Computer Sciences. http://archive.ics.uci.edu/ml.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–30.

R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org.

Sakar, C Okan, S Olcay Polat, Mete Katircioglu, and Yomi Kastro. 2019. "Real-Time Prediction of Online Shoppers' Purchasing Intention Using Multilayer Perceptron and Lstm Recurrent Neural Networks." *Neural Computing and Applications* 31 (10): 6893–6908.

Van Rossum, Guido, and Fred L. Drake. 2009. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.